REMARKS/ARGUMENTS

Claims 1, 11, and 20 are pending in the present application and have been amended. No claims were added or canceled. Support for the claim amendments can be found in the Specification, for example, on page 116, line 1 to page 117, line 12. Reconsideration of the claims is respectfully requested in view of the above amendments and the following comments.

I. 35 U.S.C. § 103, Obviousness

The Examiner has rejected claims 1, 11, and 20 under 35 U.S.C. § 103 as being unpatentable over IBM-HPM (art of record, "IBM Hardware Performance Monitor (hpm)", August 2002 (hereinafter "HPM-IBM") in view of Kosche (art of record, U.S. Patent Publication Number 2003/0005422A1) (hereinafter "Kosche") and in further view of Sato (art of record, U.S. Patent Number 6,681,388) (hereinafter "Sato"). This rejection is respectfully traversed.

In rejecting the claims, the Examiner states:

> **Claim 1:**
>
> IBM-HPM discloses a computer program product, a system, and *a method, in a data processing system, for optimizing runtime execution of a computer program* (e.g., pages 3, 8), *comprising:*
>
> *obtaining performance profile data accumulated during a trace of a computer program execution* (e.g., pages 3, 6; and page 27, obtaining and saving performance profile data accumulated in performance files),
>
> *modifying performance profile data accumulated during a trace of a computer program execution to form annotated performance profile data* (e.g., page 27, taking as input said performance files, modifying/annotating said performance profile data, and displaying said annotated performance profile data in HPMVIZ windows as in page 28),
>
> *wherein the annotated performance profile data includes annotations based on the occurrence of one or more events during execution of the computer program* (e.g., pp. 27-28, using colors, annotating corresponding source code which can be edited/optimized, displaying performance profile data and derived hardware metrics)
>
> *wherein the one or more events occur based on hardware counter values and performance indicators associated with one or more portions of the computer program* (e.g., pages 4-5);
>
> *obtaining code for the computer program (e.g., pages 27-28);*
>
> *determining a manner for compiling the code to provide one or more optimizations to the runtime execution of the computer program based on the performance profile data and the annotations of the annotated performance profile data (e.g., pages 27-29),*
>
> *wherein the one or more optimizations include at least one of an optimization to* branch prediction (e.g., pages 10, 12), and
>
> *an optimization to* cache misses and cache hit rate (e.g., pages 6, 10, 12);
>
> *presenting the one or more optimizations to a programmer for selection using one or more graphical user interfaces* (e.g., pp. 27-28: using color red to

recommend optimization as current metrics are below recommended values, using color green to indicate above threshold values, showing corresponding source code which can be edited/optimized);

receiving one or more selected optimizations of the one or more optimizations selected by the programmer (e.g., pages 27-28: adjusting/optimizing based on the red values, editing the corresponding source code which can be edited/optimized); *and*

compiling the code using the one or more selected optimizations to generate an optimized computer program (e.g., page 19, recollect performance profile data after optimizing computer program/code; pp. 21-22 and 26, recompile program/code after using one or more selected optimizations; page 27, regenerate annotated performance data based on selected optimizations; and page 32, recompile/rerun the program/code with the one or more selected optimizations).

IBM-HPM discloses profiling and optimizing branch/branch misses (p.6), branch prediction (p. 10), and branch misprediction rate (p.12) but does not explicitly discloses *one or more optimizations include at least one of an optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved.*

However, in an analogous art, Kosche further discloses *one or more optimizations include at least one of an optimization to instruction paths of the computer program at branch points such that a more contiguous execution of instructions within the computer program is achieved* (e.g., [0029]; [0034-00361; FIG. 4-6 and related text).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Kosche's teaching into IBM-HPM's teaching. One would have been motivated to do so to make operations faster, executing more operations in parallel, increase performance, and avoid pipeline stalls as suggested by Kosche (e.g., [0007-00081 and [OOI I-00201).

IBM-HPM discloses profiling and optimizing cache misses and cache hit rate (pages 3, 6, 10, and 12) but neither IBM-HPM nor Kosche explicitly discloses *one or more optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines.*

However, in an analogous art, Sato further discloses *one or more optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines* (e.g., co1.29: 1-29; col. I : 29-46; FIG. 27-29 and related text).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sato's teaching into IBM-HPM and Kosche's teaching. One would have been motivated to do so to reduce inter-cache conflict and shorten the execution time of the program as suggested by Sato (e.g., col.2: 18-22).

Claim 11:

Claim 11 is a computer program product version, which recites the same limitations as those of claim 1, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claim, they also teach all of the limitations of claim 11.

> **Claim 20:**
>
> Claim 20 is a system version, which recites the same limitations as those of claims 1 and 11, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claims, they also teach all of the limitations of claim 20.

Final Office Action dated January 30, 2008, pp. 5-8.

Claim 1, as amended herein, is as follows:

> 1. A method, in a data processing system, for optimizing runtime execution of a computer program, comprising:
>
> modifying performance profile data accumulated during a trace of a computer program execution to form annotated performance profile data, wherein the annotated performance profile data includes annotations based on the occurrence of one or more events during execution of the computer program, wherein the one or more events occur based on hardware counter values and on performance indicators associated with one or more portions of the computer program;
>
> obtaining code for the computer program;
>
> determining a manner for compiling the code to provide a plurality of optimizations to the runtime execution of the computer program based on the performance profile data and on the annotations of the annotated performance profile data, wherein the plurality of optimizations comprises an optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved, and an optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines, wherein the optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved comprises determining paths of execution that are followed more often than other paths of execution, and making a path that is executed more often at the branch contiguous with the branch instruction, and wherein the optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines comprises determining from the performance profile data that there is a problem with false cache line sharing in execution of the computer program, and arranging the instructions or data in memory in a manner that minimizes cache line sharing;
>
> presenting the plurality of optimizations to a programmer for selection using a graphical user interface;
>
> receiving one or more selected optimizations of the plurality of optimizations selected by the programmer; and
>
> compiling the code using the one or more selected optimizations to generate an optimized computer program.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka,* 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are… determined; differences between the prior art and the claims at issue are… ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). "Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). "*Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.* Id. (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006))."

In the present case, the Examiner has failed to establish a *prima facie* case of obviousness because neither IBM-HPM nor Kosche nor Sato nor their combination teaches or suggests all of the features of claim 1 as amended herein. In particular, the references, considered alone or in combination, do not teach or suggest at least the feature of "determining a manner for compiling the code to provide a plurality of optimizations to the runtime execution of the computer program based on the performance profile data and on the annotations of the annotated performance profile data, wherein the plurality of optimizations comprises an optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved, and an optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines, wherein the optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved comprises determining paths of execution that are followed more often than other paths of execution, and making a path that is executed more often at the branch contiguous with the branch instruction, and wherein the optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines comprises determining from the performance profile data that there is a problem with false cache line sharing in execution of the computer program, and arranging the instructions or data in memory in a manner that minimizes cache line sharing" as recited in amended claim 1.

Claim 1 as amended positively recites determining a manner for compiling the code to provide a plurality of optimizations to the runtime execution of the computer program based on the performance profile data and on the annotations of the annotated performance profile data, wherein the plurality of optimizations comprises an optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved, and an optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines. Neither IBM-HPM, nor Kosche nor Sato nor their combination discloses or suggests determining a manner to compile code to provide both optimizations as now recited in claim 1. Therefore, claim 1 patentably distinguishes over the references for at least this reason.

In addition, claim 1 has also been amended to recite the manner in which the two optimizations are achieved. In particular, claim 1 now recites that the optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved by determining paths of execution that are followed more often than other paths of execution, and making a path that is executed more often at the branch contiguous with the branch instruction, and that the optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines is achieved by determining from the performance profile data that there is a problem with false cache line sharing in execution of the computer program, and arranging the instructions or data in memory in a manner that minimizes cache line sharing. These features are also not disclosed or suggested in either IBM-HPM, nor Kosche nor Sato nor their combination, and claim 1 patentably distinguishes over the references for this reason as well.

For at least all the above reasons, the Examiner has not established a *prima facie* case of obviousness in rejecting claim 1, and claim 1 patentably distinguishes over IBM-HPM in view of Kosche and Sato in its present form.

Claims 11 and 20 have been amended in a similar manner as claim 1 and also patentably distinguish over the references in their present form for similar reasons as discussed above with respect to claim 1.

Therefore, the rejection of claims 1, 11, and 20 under 35 U.S.C. § 103 has been overcome.

II. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. It is, accordingly, respectfully requested that the Examiner so find and issue a Notice of Allowance in due course.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: July 30, 2008

Respectfully submitted,

/Gerald H. Glanzman/

Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants